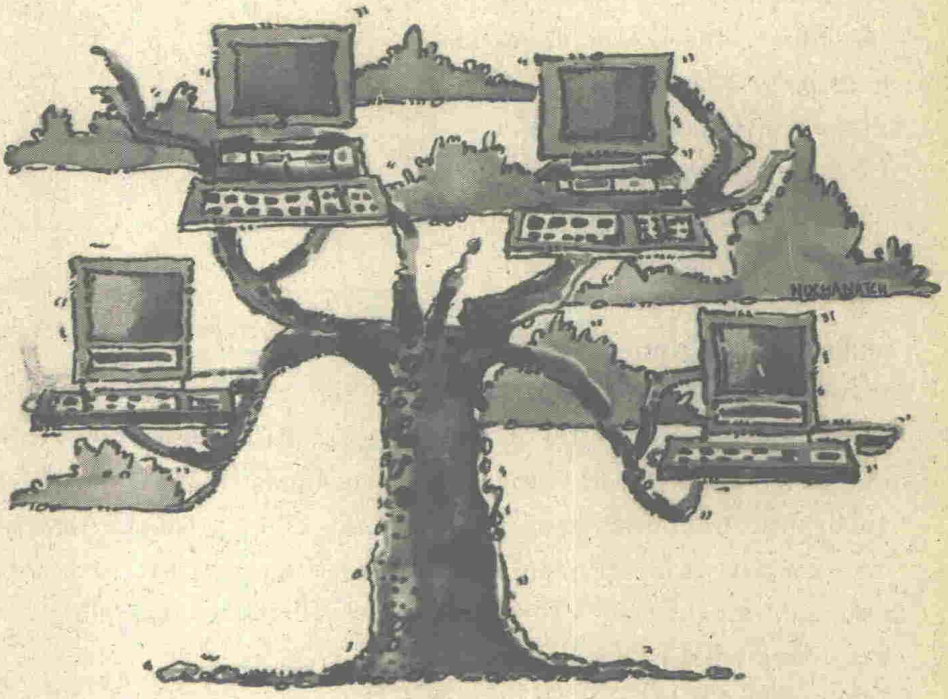
ซอฟต์แวร์

วิทยา สงวนวรรณ

เทคนิค AutoCAD : การจัดการฐานข้อมูลด้วย ASE ตอนที่ 1

เทคนิคการประยุกต์ใช้งานออโต้แคดเพื่อสามารถนำจุดเด่นของโปรแกรมออกมาใช้งานแบบเต็มประสิทธิภาพ



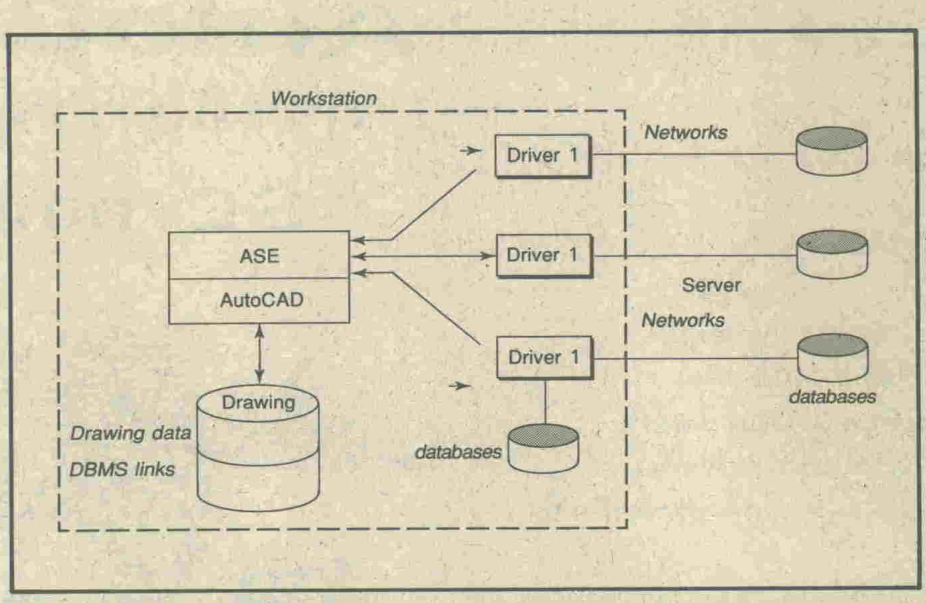
ริ่มต้นใหม่กับสิ่งที่ดีกว่า ผมขออนุญาตกล่าวคำนี้กับท่านผู้อ่าน ซึ่งท่านอาจจะสงสัยกับคำว่า "สิ่งที่ดีกว่า" ความจริงแล้วบทความประเภท *AutoCAD* นี้ก็ได้เคยวาดลวดลายบนวารสารไมโครฯ อยู่หลายตอน ก็นับได้ว่าเป็นเวลาอยู่พักหนึ่งเลยทีเดียว แต่บังเอิญท่านผู้เขียนท่านนั้นมีเหตุจำเป็นบางประการจึงได้เงียบหายไป แต่มาคราวนี้ถือได้ว่า *เป็นการเริ่มต้นใหม่* เพื่อเสนอสิ่งที่ดีกันอีกครั้ง ซึ่งก็ได้ฟิตซ้อมมาเป็นอย่างดี เพื่อจะมาทำการเบ่งกล้ามให้ท่านได้ชมพร้อมกับความมั่นใจ และก็ขอบอกได้เลยว่าคอลัมน์นี้เป็นคอลัมน์ที่ว่ากันถึง เทคนิคการประยุกต์ใช้งานออโต้แคดที่จะทำให้ท่านสามารถนำจุดเด่นของโปรแกรมออกมาเพื่อการใช้งานแบบเต็มประสิทธิภาพ และยังเปิดโอกาสให้ท่านผู้อ่านเขียนจดหมายหรือโทรศัพท์มาซักถาม อาจจะผ่านทางวารสารไมโครฯ หรือผู้เขียนโดยตรงก็ได้ ซึ่งเชื่อได้ว่าจะเป็นประโยชน์ต่อผู้ใช้งานโดยตรงมากที่สุด ส่วนท่านผู้อ่านที่เป็นมือใหม่ก็อดใจรอสักนิด โดยอีกไม่นานนักหนังสือ AutoCAD Release 12 ที่ผู้เขียนได้บรรจงสร้างขึ้นมาก็คงจะออกสู่สายตาท่านเช่นกัน

ถ้าพูดกันถึงเรื่องราวของแคดในปัจจุบันก็ต้องยกให้ *AutoCAD* อย่างแน่นอน เพราะความที่เป็นโปรแกรมยอดฮิตในแบบที่เรียกว่า "ผูกขาด" โดยสามารถกวาด Market Share เข้าไปได้ถึง 70 กว่าเปอร์เซ็นต์ ทิ้งคู่แข่งอย่างชนิดหายห่วงไปเลย และแน่นอนออโต้แคดที่พูดถึงจะติดลงท้ายด้วยเลข 12 (รีลิส 12) ที่เต็มเปี่ยมไปด้วยพละกำลังภายหลังจากที่ Autodesk Inc. ผู้สร้าง AutoCAD ได้ประเคนใส่เข้าไปสมใจกับที่ได้รอคอยมาเป็นเวลาถึง 2 ปีเต็ม นับจากรีลิส 11 เป็นต้นมา ในคราวนี้ก็ได้เตรียมยอดขุนพลระดับแนวหน้ามาไว้ให้เราอย่างสะใจ นับตั้งแต่ ASE (AutoCAD SQL Extension), AME (Advanced Modeling Extension), ADS (AutoCAD Development System), การ Render ไล่ไปจนถึงการที่ได้สร้างภาษาใหม่ขึ้นมา นั่นก็คือ DCL (Dialogue Control language) ซึ่งล้วนแต่เป็นขุมกำลังที่เย้ายวนใจน่าดึงมาใช้งานทั้งนั้น และเช่นกันก็คงเป็นหน้าที่ของผมที่จะต้องนำสิ่งเหล่านี้ไม่ว่าจะเป็นเทคนิค วิธีการใช้งาน รวมไปจนถึงสิ่งที่น่าสนใจมาตีแผ่ พร้อมตอบข้อซักถามที่เป็นประโยชน์เพื่อท่านผู้อ่านโดยเฉพาะ ซึ่งในยกแรกนี้ถือว่าเป็นการเปิดตัวของคอลัมน์ "เทคนิค AutoCAD" โดยจะเป็นการว่ากันถึง ASE กันก่อน

บริษัท TEAM INFORMATION SYSTEMS (TIS)

เริ่มนับหนึ่ง

การจัดการฐานข้อมูลหรือที่เรียกว่า *Database* บนออโต้แคด นับได้ว่าเป็นแนวความคิดที่วิเศษมาก ซึ่งแต่เดิมจะกระทำได้ในลักษณะ Attribute ที่มีข้อจำกัดอยู่หลาย ๆ เรื่อง แต่ในรีลิส 12 ได้ผนวกเอา *ASE (AutoCAD SQL Extension)* เสริมเข้าไปอีก ทำให้ท่านสามารถทำการจัดการเกี่ยวกับฐานข้อมูลได้อย่างง่ายดายบนออโต้แคด ซึ่งตามปกติแล้วออโต้แคดเป็นโปรแกรมที่เหมาะสำหรับการขีดเขียนลายเส้นให้เป็นรูปร่างชิ้นงานตามที่ต้องการเท่านั้น ยกตัวอย่างง่าย ๆ สมมติว่าท่านวาดรูปหน้าตัดของเสาต้นหนึ่งบนแบบแปลนอาคารของเรา นอกจากท่านจะทราบเพียงขนาดกว้าง×ยาวของเสานี้แล้ว ก็จะไม่อาจทราบข้อมูลที่สำคัญชนิดอื่นเลย อาทิเช่น เสาต้นนี้รับน้ำหนักได้กี่ตัน ถูกสร้างขึ้นมาไว้ที่ชั้นไหน มีจำนวนทั้งหมดกี่ต้น จะสร้างเสร็จเมื่อไร ใช้คอนกรีตขนาดกำลังเท่าไร ใช้เหล็กทั้งหมดเท่าไร หรืออื่น ๆ ที่เราต้องการจะทราบ ซึ่งท่านจะต้องไปค้นหาข้อมูลเองทั้งหมดจากหลาย ๆ แหล่งมาประกอบกัน คงจะสร้างความปวดหัวและต้องเสียเวลาไม่ใช่น้อย เมื่อได้ข้อมูลมาแล้วก็ไม่แน่ว่าจะถูกต้องด้วยหรือไม่ แต่ด้วยการอาศัยการพัฒนาให้มีการจัดการดาต้าเบสบนโปรแกรมออโต้แคด โดยผ่าน SQL จะทำให้เราสามารถสอบถามหรือค้นหาข้อมูลทุกอย่างที่ต้องการได้อย่างง่าย ๆ เพียงให้เราใช้อุปกรณ์ชี้ไป Pick แตะที่กลุ่ม Entity ที่ต้องการเท่านั้น ก็จะแสดงข้อมูลอย่างครบถ้วนบนไดอะล็อกบ็อกซ์ หรือแม้แต่จะค้นหาว่าเสาที่มีขนาดตามที่กำหนดนี้อยู่ตรงตำแหน่งใดของอาคารก็สามารถทำได้เช่นกัน นับได้ว่าเป็นก้าวหนึ่งที่สำคัญของออโต้แคดที่ทำให้ตัวมันเองเข้าใกล้การเป็น GIS (Geographic Information System) มากยิ่งขึ้น *(ซึ่งรายละเอียดจะพูดถึงอีกทีหนึ่งในฉบับต่อ ๆ ไป)*



รูปที่ 1 แสดงความสัมพันธ์ในแบบ ASE ระหว่างออโต้แคดกับดาต้าเบสภายนอก

เข้าใจ ASE อย่างง่าย ๆ

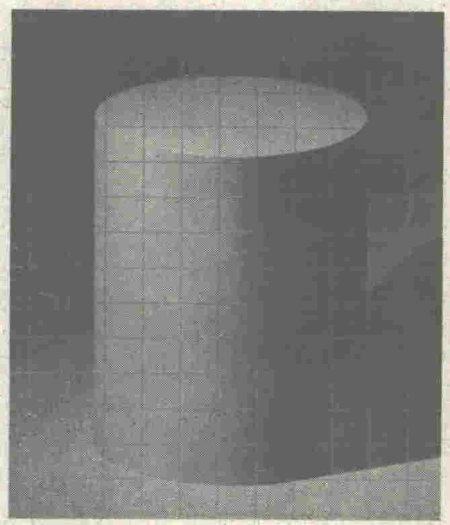
ถ้าท่านดูรูปที่ 1 ประกอบจะพบว่าเราจะมีตัวหลักที่เป็นเส้นประที่เขียนกำกับไว้ว่าเวอร์กสเตชัน แล้วมีเน็ตแวร์ต่อเชื่อมกับดาต้าเบสภายนอก ซึ่งจะหมายถึงการต่อลิงค์กับดาต้าเบสภายนอกไปได้อีกเรื่อย ๆ ลักษณะนี้เราจะถือว่าเป็น *External Database* ซึ่งความจริงแล้วเราอาจจะมองเวอร์กสเตชันเป็นพีซีตัวหนึ่งก็ได้ แล้วเอาสายจากอุปกรณ์เก็บข้อมูลต่อพ่วงเข้ามาอีกที และเจ้าดาต้าเบสที่กล่าวถึงนี้จะต้องถูกสร้างขึ้นมาจากภายนอกไว้ก่อน แล้วจึงค่อยทำการลิงค์ต่อกับออโต้แคดโดยผ่าน ASE ในภายหลัง นับได้ว่าเป็นข้อดีที่ช่วยให้ท่านสามารถสร้างดาต้าเบสขึ้นจากโปรแกรมประเภท DBMS *(Database Management System)* ที่มีความสามารถและเหมาะสมสำหรับการจัดการทางด้านดาต้าเบสมากกว่าเป็นตัวจัดการแทน และโปรแกรมประเภท "DBMS" นี้ถ้าเรียกชื่อเต็มหลายท่านอาจจะงงก็ได้ แต่ถ้าบอกว่าเป็นโปรแกรมประเภทดีเบสหรือพาราด็อกซ์ก็คงร้องอ๋อกันนะครับ

กลับเข้ามาดูในรูปที่ 1 อีกทีหนึ่ง จะพบว่าแต่ละดาต้าเบสจะต้องมีไดรเวอร์ที่เปรียบเสมือน "ตัวแปลง" ให้สามารถคุยกับออโต้แคดได้รู้เรื่อง เหตุผลก็คือดาต้าเบสของเราอาจจะถูกสร้างขึ้นจากหลาย ๆ โปรแกรมก็ได้ เช่น dBASE3, dBASE4, ORACLE, PARADOX หรือ INFORMIX ก็ยิ่งจำเป็นที่จะต้องมีไดรเวอร์มารองรับตามลักษณะฟอร์แมตของข้อมูลที่สร้างขึ้นมา ว่ากันง่าย ๆ ถ้าดาต้าเบสของเรานั้นถูกสร้างขึ้นมาจาก DBMS ใดก็ตาม ก็จะต้องมีไดรเวอร์ตามนั้นเลยทีเดียว ถึงแม้จะใช้ไดรเวอร์หลายตัว แต่ทุกตัวก็ยังคงจะต้องผ่านการควบคุมจากลูกพี่ "ASE" เสมอ

เอาละตอนนี้ท่านรู้แล้วว่าออโต้แคดมี ASE ไว้เพื่อติดต่อกับดาต้าเบสภายนอก *(External Database)* และเราจะมาดูดาต้าเบสให้ลึกซึ้งหน่อยว่ามันเป็นอะไร ความจริงแล้วดาต้าเบสหรือที่เรียกว่าฐานข้อมูลจะเป็นสิ่งที่ใช้เป็นแหล่งเก็บข้อมูลเพื่อการวิเคราะห์และประมวลผลที่ส่วนใหญ่จะเป็นตัวอักขระหรือเท็กซ์ที่เรา ๆ ท่าน ๆ สามารถอ่านกันรู้เรื่อง

แต่สิ่งที่น่ากลัวของดาต้าเบสมันอยู่ที่ว่าขนาดและจำนวนข้อมูลอันมากมายมหาศาลนี้เองที่ทำให้การค้นหาหรือการเข้าถึงข้อมูลที่ต้องการเป็นเรื่องที่ยากชะมัด ซึ่งในส่วนนี้จะต้องมีโปรแกรมทางด้าน DBMS ที่มีประสิทธิภาพจริง ๆ เท่านั้น ที่จะช่วยท่านได้ ถ้ามองในแง่ออโต้แคดเจ้าดาต้าเบสที่สร้างจากภายนอกนี้จะถูกเรียกว่า "Nongraphic Data" และในการใช้งานของเรานั้นเราจะทำการสร้าง Drawing ขึ้นมาก่อนที่เป็น Entity (แต่ละเส้น) บนจอภาพ ไม่ว่าจะเป็น Entity แบบเดี่ยวหรือแบบกลุ่มที่เป็นบล็อกก็ได้ เมื่อสร้างจนเป็นที่พอใจแล้วก็มาถึงจุดที่สำคัญที่สุดก็คือการลิงค์ (เชื่อมโยง) ระหว่าง Entity ที่เป็น *"Graphic data"* ผ่านคำสั่ง ASE ไปเชื่อมต่อกับ *"Nongraphic Data"* ที่เป็นดาต้าเบสของเราในลักษณะมีความสัมพันธ์ซึ่งกันและกัน หรือที่เรียกกันว่า *Relationship* นั่นเอง ก็เป็นอันเสร็จวิธีการ

เมื่อถึงตอนนี้ท่านคงพอจะมองออกนะครับว่า เพราะการมี Relationship นี้เอง ทำให้เราสามารถค้นหาข้อมูลบนออโต้แคดแทนที่จะต้องไปค้นหากันบนดาต้าเบส ที่นอกจากจะมากมายแล้วยังมองไม่เห็นผลอย่างชัดเจนได้เหมือนกับการแสดงแบบกราฟิก เมื่อ



มองดูภาพรวม ๆ แล้วก็จะคล้าย ๆ กับเป็นการรวมดาต้าเบสกับแคดให้มารวมกันนั่นเอง

จากวิธีการดังกล่าวนี้จะเป็นการเปิดโอกาสให้ท่านสามารถทำการสอบถามหรือค้นหาข้อมูลบน Drawing ของท่านได้อย่างง่ายดาย ดังนั้นการทำ BOM *(Bill of Material)* หรือการทำ Estimate *(ประมาณราคา)* ก็ไม่ใช่เรื่องที่ยากกันอีกต่อไปอีกแล้ว ซึ่งเรื่องทั้งสองนี้คงทำให้ใครต่อใครพากันขยาดมานักต่อนัก แถมยังทำได้อย่างแม่นยำถูกต้องอีกต่างหาก ผมขอพาท่านวกกลับมาสู่รูปที่ 1 อีกครั้ง ตรงบริเวณด้านล่างของรูปจะหมายถึง ลักษณะการเก็บข้อมูลของ Drawing file และที่เห็นแบ่งเป็นสองส่วนนั้นก็จะหมายถึงลักษณะการเก็บข้อมูลของออโต้แคดที่ส่วนหนึ่งจะเป็นการเก็บข้อมูลแบบปกติที่เป็น Vector ของ Entity ทั้งหมด และอีกส่วนหนึ่งก็จะเป็นส่วนที่เรียกว่า *DBMS Link* ซึ่งจะเป็นส่วนที่เก็บข้อมูลของความสัมพันธ์ระหว่างดาต้าเบสภายนอกกับ Entity (แต่มิได้หมายความว่าจะเป็นการเก็บดาต้าเบสเอาไว้ทั้งหมด ซึ่งดาต้าเบสก็ยังคงอยู่ข้างนอก Drawing เช่นเดิมอยู่) ซึ่งก็เพียงจะทำให้ขนาดข้อมูลมีขนาดใหญ่ขึ้นอีกเพียงเล็กน้อยเท่านั้น ก็ไม่หนักหนาอะไรมากนัก และถ้าเราสังเกตให้ดีอีกจะพบว่า ทิศทางของหัวลูกศรจะมีทิศทางไปกลับ นั่นก็จะหมายถึง การส่งข้อมูลไป-กลับได้ ทำให้เราสามารถแก้ไขเพิ่มเติมข้อมูลจากออโต้แคดหรือ DBMS อย่างใดอย่างหนึ่งได้อย่างสบาย ๆ เสร็จแล้วก็จะปรับแก้ตามกันไปอย่างอัตโนมัติด้วยเช่นกัน

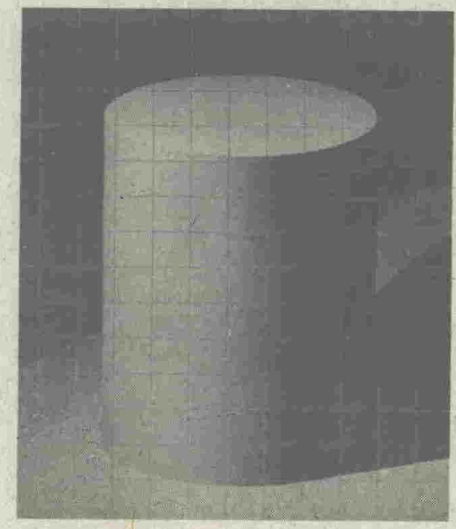
สิ่งที่ผมได้ยกมากล่าวอ้างทั้งหมดนี้ก็เป็นเพียงวิธีการคร่าว ๆ เท่านั้น ซึ่งความจริงแล้วเรื่องของดาต้าเบสและแคดนี้เป็นเรื่องที่ซับซ้อนและละเอียดอ่อนมาก อาจจะทำให้ท่านมองภาพได้ไม่ชัดเจนนัก แต่ถ้าหากท่านทำตามขั้นตอนต่าง ๆ ตั้งแต่แรกไปจนกระทั่งจบแล้วผมเชื่อว่าคงจะทำให้ท่านสามารถสั่งการบน ASE ได้เลยทีเดียว เมื่อถึงตอนนี้จะเป็นการดีมากสำหรับท่าน ถ้าท่านมีความรู้ทางด้านดาต้าเบสบ้างเล็กน้อย เอาแค่ให้เข้าใจคอนเซ็ปต์ใหญ่ ๆ ของดาต้าเบสก็พอแล้ว ดังนั้นท่านควรจะไปศึกษาพื้นฐานไว้บ้างจากหนังสือเล่มที่เกี่ยวกับดาต้าเบส เช่น dBASE, PARADOX ฯลฯ แต่ผู้เขียนขอแนะนำว่าการที่เราทำการศึกษาคอนเซ็ปต์พื้นฐานจาก *"dBASE3"* จะให้แนวคิดทางด้านดาต้าเบสที่ค่อนข้างดีและดาต้าเบสที่อ้างถึงในที่นี้ทั้งหมด ผมขออนุญาตใช้ดาต้าเบสที่ถูกสร้างขึ้นมาจาก DBMS ที่ชื่อว่า dBASE3 ส่วนโปรแกรม DBMS อื่น ๆ ก็คงจะยังอาศัยหลักการและวิธีการเดียวกันทุกประการครับ

จุดดีที่ดีจริง ๆ

ท้าวความเดิมกันบ้างสักเล็กน้อยในลักษณะที่พูดกันถึงการลิงค์ดาต้าเบสแบบลักษณะ External ในรีลิสก่อน ๆ

(ตั้งแต่รีลิส 11 ย้อนขึ้นไปเลย) ออโต้แคดจะอำนวยความสะดวกให้เราในการเชื่อมโยงข้อมูลที่เป็น *"Nongraphic Data"* หรือเท็กซ์เข้ากับ Entity ในลักษณะที่เรียกว่า Attribute ซึ่งกระทำได้แบบฝืด ๆ แถมยังไม่ค่อยจะสะดวกเอาเสียมาก ๆ เลย ก็เพราะเนื่องจากออโต้แคดไม่ได้เป็นโปรแกรมประเภท DBMS แต่ถูกออกแบบมาให้เหมาะสำหรับการขีด ๆ เขียน ๆ เท่านั้น ก็ยังดี Attribute เอาไว้พอแก้ขัดไปได้บ้าง แต่เมื่อทำไปทำมามาก ๆ เข้า Drawing ของเราชักจะโตขึ้นไปเรื่อย ๆ จนทำท่าทางอุ้ยอ้าย กว่าจะขยับไปทำอะไรสักอย่างหนึ่งก็ต้องเสียเวลา ก็ขอบอกได้เลยว่า จุดดีของการมี ASE ประการแรกที่เข้าท่าที่สุดก็คงเป็นการทำให้ข้อมูล Drawing ของท่านที่ไม่ตระกลามกินเนื้อที่ในฮาร์ดดิสค์มากจนเกินความจำเป็น มีขนาดที่สมน้ำสมเนื้อ ก็ด้วยการที่แยกข้อมูลที่เป็น Vector และข้อมูลดาต้าเบสให้แยกออกจากกันอย่างเด็ดขาดนั่นเองที่ทำให้เราสามารถพ่วงต่อดาต้าเบสเข้าไปได้อีกเรื่อย ๆ และก็ไม่จำเป็นที่จะต้องทำให้ขนาด drawing file ใหญ่โตตามไปด้วย เผลอ ๆ ดาต้าเบสของเราอาจจะมีขนาดมหาศาลจนติดต่ออุปกรณ์จัดเก็บข้อมูลเพิ่มเข้าไปอีกก็เป็นได้ นอกจากนี้ถ้าเกิดเราต้องการเฉพาะรูป drawing ของเราไปพล็อตออกที่อื่น ๆ หรือที่ไหนๆ ก็สามารถทำได้โดยไม่ต้องพ่วงหอบดาต้าเบสไปให้หนักเปล่า ๆ

ข้อดีข้อหนึ่งก็คือ การแก้ไขเพิ่มเติมหรือสร้างดาต้าเบสสามารถกระทำได้สะดวกแบบสบาย ๆ ก็เพราะการที่ถูกแยก *Database* ออกจาก *Vector* นี้เอง เป็นการเปิดโอกาสให้ทำอะไรก็ได้บนดาต้าเบสของเรา ซึ่งสามารถใช้โปรแกรมประเภท DBMS ที่เป็นมือ-



อาชีพทางด้านการจัดการดาต้าเบสได้อย่างเหมาะสมกว่า ไม่ต้องผ่านออโต้แคดให้รำคาญใจ เมื่อแก้ไขแล้วก็จะทำการอัพเดตข้อมูลใน drawing ที่เราเคยลิงค์ไปแล้วอย่างอัตโนมัติ หรือถ้ายังไม่สะใจจะไปแก้ไขดาต้าเบสบนออโต้แคดโดยตรงเลยก็สามารถทำได้ เมื่อเสร็จแล้วจะไปอัพเดตที่ดาต้าเบสอีกเช่นกันสะดวกแบบสบายจริง ๆ

ข้อดีข้อสุดท้ายก็คือ การที่สามารถมีชุดดาต้าเบสได้หลายชุดและสามารถเลือกใช้ดาต้าเบสในชุดที่ต้องการจะใช้งานได้ ซึ่งข้อดีข้อนี้จะช่วยให้การทำงานเป็นระบบและเป็นขั้นเป็นตอน *ง่ายต่อการทำงานและตรวจแก้จุดที่ผิดพลาด* โดยความเป็นจริงแล้วการใช้งาน ASE อาจจะไม่จำเป็นจะต้องลิงค์กับดาต้าเบสทุกตัวก็ได้ เราอาจจะจัดกลุ่มให้เหมาะสมสัก 2-3 กลุ่ม แล้วตั้งชื่อตามความสัมพันธ์ที่สร้างขึ้น เมื่อต้องการจะใช้ชุดใดก็แค่เปลี่ยนชื่ออ้างอิงเท่านั้น ทำให้สะดวกต่อการใช้งานและจะได้ใช้ดาต้าเบสแบบเต็มประสิทธิภาพไปเลย

การทำงานลักษณะที่เป็นการแยกดาต้าเบสและแคดให้ออกจากกันนี้ถือเป็นการทำงานที่ชาญฉลาด เพราะแต่ละโปรแกรมมีความเชี่ยวชาญไปคนละแบบเรียกว่าเก่งไปกันคนละทาง แล้วจึงเอาข้อดีของทั้งสองมารวมกัน เกิดเป็นผลลัพธ์ให้เราได้เก็บเกี่ยวใช้งานได้

เข้าใจให้ตรงกันก่อน

เรื่องของดาต้าเบสเป็นเรื่องซับซ้อน ท่านต้องใจเย็น ๆ ในการทำความเข้าใจ ก็อาจจะไม่หมูเท่าไร แต่ก็คงไม่เกินมืออาชีพแบบท่านไปได้ใช่ไหมครับ! เอาละก่อนที่จะเข้าไปสู่รายละเอียด ท่านกับผมจะต้องมา lecture และทำความเข้าใจบางเรื่องให้ตรงกันก่อน เรื่องแรกก็คือ เรื่องการจัดการระบบข้อมูลที่จะต้องมีการเลือกใช้ DBMS (Database Management System) นี้จะต้องมีการพิจารณาหลายด้านประกอบกันคือ

- *ความปลอดภัยของการเก็บข้อมูล* (Data security) ควรจะใช้โปรแกรมที่ป้องกันมิให้ผู้ใช้ผู้อื่นมารบกวนหรือแก้ไขข้อมูล โดยเจตนาหรือไม่เจตนาก็ตาม

- ควรมีลักษณะของ *Relational Database* อย่างสมบูรณ์ คือ มีลักษณะการเรียกหาข้อมูลแบบ SQL *(Stuctural Query Language)* และมีกระบวนการทำงานในแบบที่เรียกว่า "Non-Procedural" โดยมีหลักการว่าหากต้องการข้อมูลอะไร เพียงแต่บอกไปว่าต้องการอะไรบ้าง มิใช่ว่าต้องหาวิธีการที่จะเอาข้อมูลนั้น ๆ ออกมา

- *มีลักษณะการทำงานที่รวดเร็ว* ในการค้นหาหรือการเข้าถึงข้อมูล และสามารถเชื่อมต่อกับระบบอื่นได้

ซึ่งก็เป็นเหตุผลว่าทำไมออโต้แคดจึงต้องเลือกใช้ SQL ในการจัดการดาต้าเบส และเหตุผลทั้ง 3 ประการนั้นก็มีอยู่ครบถ้วนใน ASE อยู่แล้ว ก็คงเป็นหน้าที่ของท่านที่จะเลือกใช้โปรแกรม DBMS ให้เหมาะสมด้วย

เมื่อท่านเลือก DBMS ได้แล้ว ต่อไปก็คงจะเป็นเรื่องของดาต้าเบส และดาต้าเบสที่กล่าวถึงนี้ออโต้แคดจะมองในลักษณะที่ค่อนข้างแปลกกว่าคอนเซ็ปต์ทั่ว ๆ ไปคือ ดาต้าเบสจะถูกมองในลักษณะที่เป็นกลุ่ม Tables (ดูรูปที่ 2 ประกอบ) ซึ่งถ้าสมมติว่าท่านเลือกใช้โปรแกรม DBMS เช่น dBASE3 ในการสร้างดาต้าเบสไฟล์ขึ้นมา และไฟล์เหล่านี้จะมีนามสกุล ".DBF" ถ้าให้มีชื่อดังนี้ : invantry.dbf, Computer.dbf และ Employee.dbf และถ้าตั้งชื่อดาต้าเบสกลุ่มนี้ว่า "TUTOR"

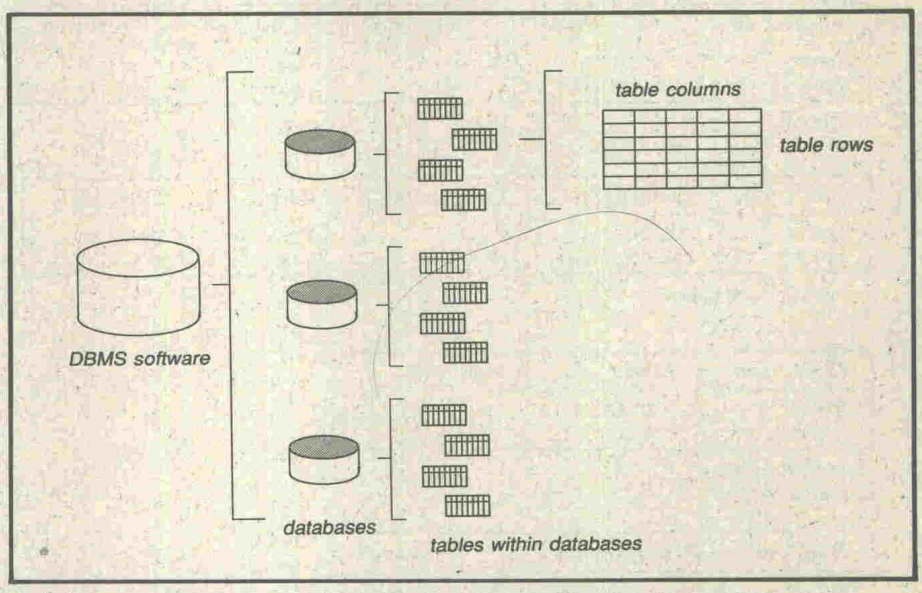
ดังนั้นถ้าพูดถึง *Table* ก็จะหมายถึงทั้ง 3 ไฟล์ที่มีนามสกุล .DBF และถ้าพูดถึง*ดาต้าเบส* ก็จะหมายถึงกลุ่มของ Table ที่ประกอบไปด้วย Table ของ Inventry, Table ของ Computer และ Table ของ Employee นั่นเอง และที่สำคัญมากก็คือ ในทุก ๆ table ที่ประกอบกันเป็นดาต้าเบสนี้จะต้องมีความสำคัญในลักษณะซึ่งกันและกัน ที่เรียกว่า "Relational Database" อีกด้วย (จะกล่าวถึงอีกทีหนึ่ง)

คราวนี้เราลองมาพิจารณา Table ของเราว่าแต่ละส่วนยังมีส่วนประกอบอะไรบ้างอีก (ดูตารางที่ 1.1 ประกอบ) Table ที่เราเห็นในตารางที่ 1 จะเป็นสิ่งที่ใช้เรียกแทนไฟล์ที่ถูกสร้างขึ้นมาจากโปรแกรม DBMS ที่ชื่อว่า dBASE3 จนได้นามสกุล ".DBF" ซึ่งอาจจะดูขัด ๆ กับความรู้สึกไปบ้างก็ตาม ซึ่งดาต้าเบสกลับหมายถึง *ทั้งกลุ่ม Table แทน* ไม่ได้หมายความถึงแต่ละไฟล์ที่ถูกสร้างขึ้นจาก DBMS และ Table นี้ยังแบ่งออกเป็น Rows และ Columns ได้อีก

- Rows ก็คือจำนวนแถวในแนวนอน ซึ่งจะบอกรายละเอียดของข้อมูลที่เก็บ ซึ่งส่วนนี้เองแหละที่ใช้ลิงค์กับ Entry Rows นี้อาจจะเรียกว่า *Record* ก็ได้

- Columns ก็คือ หลักในแนวตั้งของตารางที่เป็นตัวบอกข้อมูล ที่เราจัดเก็บไว้นั่นเองว่าเป็นอะไรบ้าง และคอลัมน์นี้อาจจะเรียกว่า *Filed* ก็ได้ จึงทำให้ตารางที่ 1 นี้ประกอบไปด้วย 1 Table, 5 Field และ 6 Record

ถึงตรงนี้ท่านก็คงพอจะคุ้นเคยกับความหมายของ Database, Table, Record และ Field กันบ้างนะครับ ก็ยังมีอีกสิ่งหนึ่งที่สำคัญจนมิอาจจะละเลยได้ นั่นก็คือ การสร้างความสัมพันธ์ซึ่งกันและกัน (Relational Database) ซึ่งจะเป็นตัวจักรสำคัญที่ใช้บ่งบอกถึงประสิทธิภาพการเข้าถึงข้อมูลของดาต้าเบส ซึ่งจะคล้าย ๆ กับการเชื่อมต่อแต่ละ Table เข้าด้วยกัน โดยอาศัยการส่งผ่าน "Field" *ที่เหมือนกัน* นั่นเอง ดังแสดงในตารางที่ 2, 3, 4 และ 5 (ที่เป็นลูกศรชี้แสดง)



รูปที่ 2 แสดงลักษณะโครงสร้างและความสัมพันธ์ของดาต้าเบส

ตารางที่ 1 แสดงถึงดาต้าเบสส่วนที่เป็น Table

Column5 Table

NAME	EDUCATION	AGE	POSITION	SALARY
WINAI T.	DIPLOMA	22	DRAFTMAN	7,000
SOMCHOK L.	DIPLOMA	23	DRAFTMAN	9,000
RATEE S.	MASTER D.	35	MANAGER	3,500
NOPPADON CH.	BACHELOR D.	30	SUPERVISOR	20,000
CHINTANA L.	DIPLOMA	25	TECHNICIAN	12,000
SARAWUT M.	DIPLOMA	27	TECHNICIAN	15,000

อะไรคือ SQL?

SQL นี้ไม่ใช่เรื่องใหม่อะไรกันเลย ความจริงเขามีการพัฒนากันมานานแล้ว จนกลายเป็นภาษาอีกภาษาหนึ่งที่ใช้ในการค้นหาและสอบถามข้อมูลที่จะต้องมีการว่ากันอีกยาวเลยทีเดียว

Order_no	Sale_date	Staff_no	Cust_no	Invoiced
020002	09/21/87	000008	000025	F
020006	09/22/87	000012	000036	F

ตารางที่ 2 Sales

Cust_no	Company	Last_name	Address	Dity, ST
000001	Fine Designs	Leonard	155 Bayfront	Ojai, CA
000027	Classic Interiors	Erickson	200 Main St.	St.Louis, MO

ตารางที่ 3 customer

Order_no	Part_no	Shipped
020002	001025	F
020006	001007	F

ตารางที่ 4 Items

Part_no	Description	Location
001007	Wodd desk, single pedestal	New York
001025	Desk, executive, 5 ft.	Los Angles

ตารางที่ 5 Inventory

แต่เดิมถ้าเรามองออโต้แคดในทางด้านการจัดการที่เกี่ยวกับดาต้าเบส ซึ่งปกติจะทำได้ค่อนข้างแย่ แต่เมื่อมิอาจขยับตัวมาทางด้านดาต้าเบสที่เป็น External ก็จำเป็นจะต้องอาศัยภาษาตัวนี้เข้ามาเป็นส่วนช่วยเสริมให้ตัวมันเอง "แข็งขึ้น" SQL นี้จะออกเสียง "ซีเควล" หรือ "S-Q-L" ก็ได้ ซึ่งจะย่อมาจาก *"Structured Query Language"* ที่เหมาะสมสำหรับการสอบถามหรือค้นหาข้อมูลที่เป็นโครงสร้าง โดยตัวมันเองจะเป็นภาษาที่ใช้ในการปฏิบัติงานและควบคุมกับฐานข้อมูลสัมพันธ์ (Relational Database) ซึ่ง SQL นี้จะเป็นภาษาที่เป็นภาษามาตรฐานที่นิยมใช้งานได้โดยทั่ว ๆ ไป และเป็นภาษาที่ค่อนข้างเรียนรู้ได้ง่ายเมื่อเทียบกับภาษาโครงสร้างตัวอื่น แม้แต่ผู้ใช้งานที่ขาดประสบการณ์ในการประมวลผลข้อมูลก็สามารถที่จะเรียนรู้ภาษานี้ได้อย่างรวดเร็ว

SQL นี้จะทำให้เราสามารถติดต่อกับ Table ได้โดยตรงในขณะที่กำลังทำงานอยู่บนออโต้แคด ซึ่งทำให้เราควบคุมการจัดการดาต้าเบสแบบอยู่ในมือเลยทีเดียว ความจริงคำสั่งใน SQL จะมีอยู่มากเอาการ แต่ถ้าใช้งานจริง ๆ แล้วจะมีอยู่ไม่กี่ตัว นอกนั้นจะเป็นตัวเสริมในการแยกแยะข้อมูลให้ดูดีขึ้นเท่านั้นเอง ตอนนี้ผมจะขอพับเก็บเรื่องราวของ SQL เอาไว้ก่อน เอาไว้ถึงขั้นตอนการประยุกต์ใช้งานก็จะกล่าวอย่างละเอียดอีกทีหนึ่ง เดี๋ยวท่านจะเบื่อเสียก่อนนะครับ

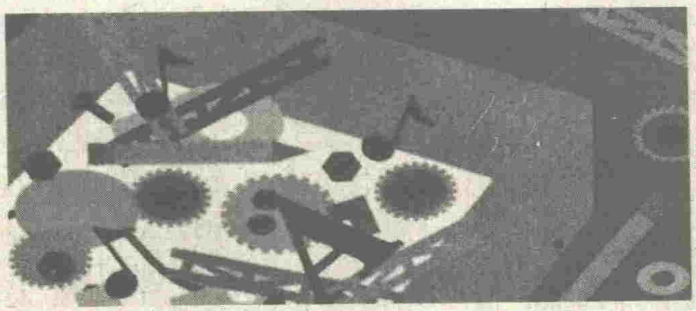
เริ่มต้นกันเสียที

เอาล่ะ ตอนนี้จะเข้าเรื่องจริง ๆ เสียที หลังจากที่พาท่านไปทัวร์ดูรอบ ๆ ASE จนพอจะมองออกว่ารูปร่างหน้าตาเป็นอย่างไร ตอนนี้ผมจะพูดถึงวิธีการประยุกต์ใช้งานโดยคร่าว ๆ ก่อนเพื่อให้ท่านได้ทราบแนวทางการทำงานไว้ก่อนดังนี้

สิ่งแรกที่ท่านจะต้องทำก็คือ การตั้งค่าระบบ*คอนฟิกูเรชัน*ของดาต้าเบสขึ้นก่อนขณะที่อยู่ในดอสเพื่อให้ออโต้แคดสามารถค้นหาและอ้างอิงกลุ่ม Tables ที่เรากำหนดได้ ซึ่งจะกระทำให้ลักษณะการเซตพาธของดาต้าเบสขณะที่ทำงานอยู่บนดอส (ก่อนเข้าออโต้แคด) หรือถ้าจะให้สะดวกจะนำไปใส่ในแบตช์ไฟล์ "ACADR12.BAT" เลยก็ได้ ถัดจากนั้นมาก็เข้าสู่ออโต้แคด เปิดไฟล์ drawing ที่เราต้องการในแบบปกติขึ้นมาใช้งาน จากนั้นก็ทำการ Initialize ASE เพื่อเตรียมใช้งานบนออโต้แคด ในขั้นตอนนี้ถ้าสมมติว่าเรามีการลิงค์กับดาต้าเบสไปแล้ว ออโต้แคดก็จะถามถึง Username และ Password เพื่อเป็นการปกป้องข้อมูลของท่านไม่ให้ใครเข้าไปยุ่มย่ามกับข้อมูลโดยไม่จำเป็น จากนั้นก็ขยับไปใช้คำสั่งใน ASE เพื่อทำการลิงค์ "Records" หรือ rows ใน Table ให้เชื่อมต่อกับ Entity บน drawing ในหน้าจอของเรา ซึ่งข้อมูลในดาต้าเบสหลังจากที่ได้ลิงค์ไปแล้วก็จะยังสามารถเรียกใช้งานหรือแก้ไขได้ตามปกติ

ยังมีสิ่งหนึ่งที่สำคัญมากก็คือ ออโต้แคดจะนำข้อมูลในส่วนที่ลิงค์กันทั้งหมดจัดเก็บไว้ใน drawing file ในlayer ที่ชื่อว่า *ACADASE* ซึ่งเราจะต้องไม่ไปรบกวนใน layer นี้เด็ดขาด มิฉะนั้นข้อมูลที่ลิงค์กันไปแล้วจะเสียหายได้ และหลังจากที่ท่านลิงค์ดาต้าเบส (ที่เป็นเรคอร์ด) เชื่อมโยงกับ Entity จนเป็นที่ต้องการทั้งหมดแล้ว ก็คือว่าสมบูรณ์พร้อมที่จะเรียกใช้งานบนภาษา SQL ได้ ถึงตอนนี้ท่านจะทำ BOM (Bill of Material) หรือ Estimate ย่อมทำได้อยู่แล้ว ใช่ไหมครับ

เพื่อให้ท่านสามารถเข้าใจได้อย่างรวดเร็ว ผมจะขออนุญาตใช้ drawing



รูปที่ 3

```
C:\ >                                    <ขณะอยู่ที่ C prompt>
C:\ > CD ACAD12                          <ไปยังไดเรคทอรีที่ ACAD อยู่>
C:\ > ACAD12> MD TEMP                    <สร้าง sub ขึ้นมาใหม่ชื่อ TEMP>
C:\ > ACAD12> CD TEMP                    <ไปยังไดเรคทอรี TEMP>
C:\ > ACAD12\TEMP>COPY \ACAD12\TUTORIAL\ASETUT.*
C:\ > ACAD12\TEMP>COPY \ACAD12\TUTORIAL\DBF\*.*
```

รูปที่ 4

```
ASETUT      DWG     89882    bytes
INVENTRY    DBF     34891    bytes
EMPLOYEE    DBF     3483     bytes
COMPUTER    DBF     667      bytes
```

file และดาต้าเบสที่ทางออโต้แคดได้เตรียมไว้ให้ท่านอยู่แล้วใน Tutorial เพื่อที่ท่านจะได้ไม่ต้องเหนื่อยกับการสร้างดาต้าเบส และท่านก็สามารถทดลองทำตามได้เลยทันที แต่ขอบอกก่อนได้เลยว่า drawing file ตัวนี้ (ชื่อ ASETUT) ได้มีการลิงค์ Entity กับดาต้าเบสไว้บ้างแล้ว แต่ยังไม่สมบูรณ์ ถึงอย่างไรผมจะแสดงทุกขั้นตอนใหม่อย่างละเอียดเลยนะครับ และเพื่อให้ง่ายต่อความเข้าใจ ผมจึงขอแบ่งออกเป็นหัวข้อย่อย ๆ ได้ดังนี้

1. การเตรียมการก่อนใช้งาน (Database-Specific Configuration)
2. สร้างฐานข้อมูลด้วย dBASE3
3. การเริ่มใช้งาน ASE (Initialization)
4. การสร้าง link Entity
5. การสร้าง Attribute
6. การแก้ไข Database บน AutoCAD
7. การใช้งานคำสั่ง SQL
8. การสร้างรายงาน
9. สิ้นสุดการใช้งาน ASE

เตรียมตัวกันก่อน

ก็เริ่มกันที่หัวข้อที่ 1 กันเลย ในขั้นตอนแรกนี้จะเป็นการเตรียมการกันก่อน ซึ่งก่อนจะใช้งานบน ASE ทุกครั้งจะต้องมาทำการตั้งค่าคอนฟิกูเรชันของดาต้าเบสขณะอยู่บนดอสเสมอ ไม่เช่นนั้นออโต้แคดจะหาค่า Table ต่าง ๆ ไม่พบเดี๋ยวจะพาลทำงานบน ASE ไม่ได้เสียดื้อ ๆ ไปเลย และเพื่อความสะดวกเราจะจัดการเก็บค่านี้ไว้ในแบตช์ไฟล์เสียเลยไม่ว่าจะเป็น Autoexec.bat หรือ ACADR12.bat ก็ได้ แต่ก่อนลงมือทำผมขอสำรองไฟล์ไว้ที่จะใช้งานก่อนให้ปลอดภัยจากความผิดพลาด เรียกว่า "กันเหนียว" ไว้ก่อน ดังแสดงในรูปที่ 3

ถ้าท่านสร้างไดเรคทอรีของออโต้แคดไว้ที่ไดรฟ์หรือชื่ออื่น ก็อย่าลืมเปลี่ยน "ACAD12" และ "C:" ให้ถูกต้องด้วยนะครับ ในตอนนี้ถ้าเราใช้คำสั่ง DIR เข้าไปดูรายชื่อของไฟล์ก็จะพบว่ามีทั้งหมดอยู่ 4 ไฟล์ ดังแสดงในรูปที่ 4

ขั้นตอนที่สำคัญจะอยู่ที่การตั้งชื่อดาต้าเบสที่จะนำมาใช้งาน คล้ายกับเป็นการรวม Tables หลาย ๆ อันเข้าด้วยกันแล้วกำหนดชื่อลงไป การตั้งชื่อของดาต้าเบสนี้เราจะเรียกว่า การคอนฟิกูเรชันดาต้าเบส ซึ่งจะเป็นเพียงการตั้งชื่อสมมติ (Alias) ขณะอยู่ในดอส จุดประสงค์หลักก็เพื่อต้องการให้ออโต้แคดรู้ว่าท่านกำลังอ้างถึง Tables กลุ่มใดอยู่เพื่อดึงมาใช้งานเท่านั้น ตอนนี้เราได้รวบรวม Table (นามสกุล .DBF) ตั้ง 3 อัน โดยจะกำหนดชื่อดาต้าเบสลงไปว่า "ASETUT" ด้วยการพิมพ์คำสั่งนี้ขณะอยู่ในเครื่องหมาย prompt ของดอส (หรือนำไปแทรกที่แบตช์ไฟล์) ดังนี้

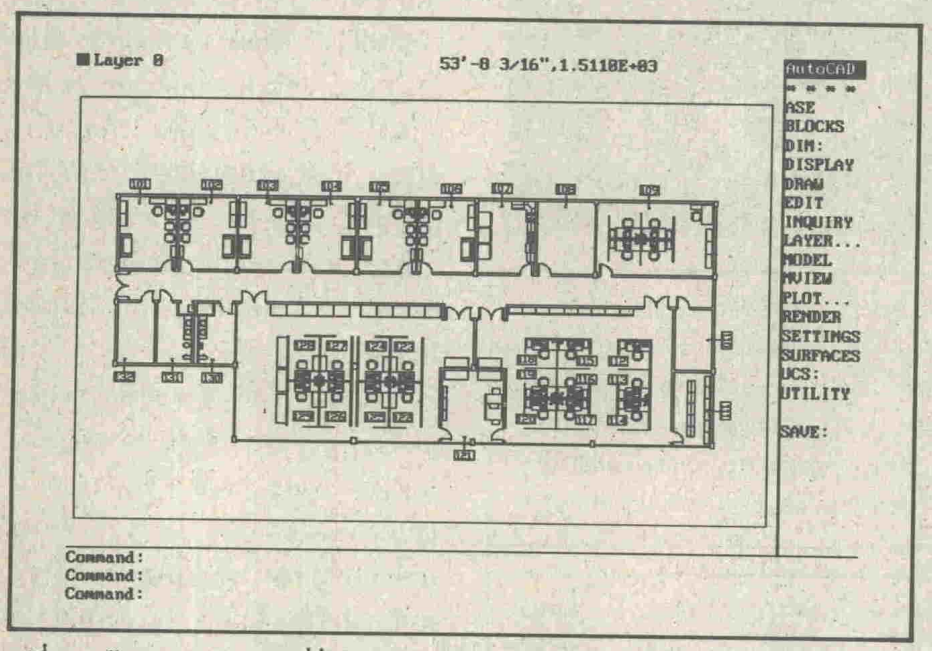
```
C:\ > SET ASETUT=C:\ACAD12\TEMP
```

จากนี้ไปออโต้แคดก็จะเข้าใจว่า

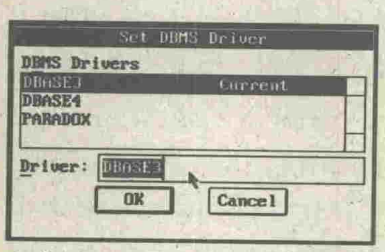
- *Table* ก็จะหมายถึง computer.dbf, Employee.dbf และ inventry.dbf
- *Database* ก็จะหมายถึง ชื่อสมมติที่เรากำหนดขึ้นที่ชื่อว่า "ASETUT"

ติดเครื่อง ASE...

เพื่อให้เร้าใจยิ่งขึ้น ผมขอพาท่านกระโจนใส่ขั้นตอนที่ 3 เลย หลังจากที่เผาหัวจนเครื่องร้อนเต็มที่แล้ว (ขั้นตอน



รูปที่ 5 หน้าตาของ Drawing ที่ชื่อว่า "ASETUT.DWG"



รูปที่ 6 แสดงไดอะล็อกบ็อกซ์ของการตั้งไดรเวอร์ของ DBMS

ที่ 2 จะนำไปกล่าวถึงในขั้นตอนที่ 4 นะครับ) ก็เริ่มจากเปิด file drawing ที่เราได้เตรียมไว้คือ "C:\ACAD12\TEMP\ASETUT.DWG" ซึ่งรูปร่างหน้าตาก็ควรจะเหมือนกับรูปที่ 5 จากนั้นก็จัดการพิมพ์คำสั่งลงไปในบริเวณ Command prompt

```
Command:ASEINIT
```

หลังจากกด Enter ออโต้แคดจะทำการ Initialization (การเริ่มใช้งาน) เพื่อโหลดแอพพลิเคชันที่ชื่อว่า "ASE" เข้าสู่หน่วยความจำ ซึ่งจะทำให้หยุดนิ่งไประยะเวลาหนึ่ง (เหมือนกับจะนึกอะไรสักอย่าง) ก็จะเป็นการเก็บคำสั่งทั้งหมดของ ASE เพื่อให้ท่านสามารถเรียกใช้งานได้ การ Initialize นี้จะเป็นการสร้างหรือเรียก *Control Database* ที่มีหน้าที่ในการเก็บข้อมูลที่เป็นความสัมพันธ์ระหว่าง DBMS กับ Table และ Entity กับ Record ขึ้นมาให้ท่านเรียกใช้งาน

ยังมีสิ่งที่น่าสนใจอีกประการหนึ่งก็คือ เมื่อทำการ Initialize ของ drawing ที่มีการลิงค์ Entity และตั้งไดรเวอร์ DBMS ไปเรียบร้อยแล้ว ออโต้แคดดูเหมือนรู้ว่ากำลังมีผู้ใช้งานผู้อื่นเข้ามายุ่มย่ามกับดาต้าเบส จะรีบปกป้องข้อมูลด้วยการถาม Username และ Password ทันที ซึ่งถ้าเกิดได้ทำการตั้งค่าไว้แล้วท่านเกิดจำ Password ไม่ได้ก็เสร็จกันพอดี

ก่อนที่จะก้าวต่อไปข้างหน้าจะต้องทำข้อตกลงกันไว้ก่อนจะได้ไปในทิศทางที่ราบรื่น เรื่องแรกก็คือ ขณะที่ทำงานอยู่บน ASE ห้ามไปแตะต้อง Layer "ACADASE" โดยเด็ดขาดเลยนะครับ เดี๋ยวข้อมูลที่เราลิงค์ไว้แล้วอาจจะเสียหายได้ เรื่องที่สองก็คือ การคอนโทรลไดอะล็อกบ็อกซ์จะต้องตั้งค่าตัวแปร "CMDDIA" ให้เป็น "1" ด้วย โดยการพิมพ์คำสั่งนี้ลงไป

```
Command: CMDDIA <ตั้งค่าให้เป็น 1>
```

มิฉะนั้นจะไม่เกิดไดอะล็อกบ็อกซ์ขึ้น อาจจะทำให้ท่านหงุดหงิดก็เป็นได้

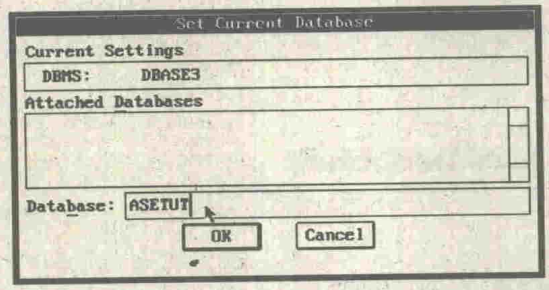
ไปโลด...

หลังจากที่ท่านติดเครื่อง ASE ด้วยการ Initialize ไปแล้ว ก็จะมาว่ากันต่อถึงขั้นตอนการทำงานในขั้นตอนที่ 3 กันต่ออย่างละเอียดอีกครั้ง โดยผมจะแบ่งเป็นขั้นตอนให้ง่ายต่อการทำตามแบบ step by step เลยนะครับ

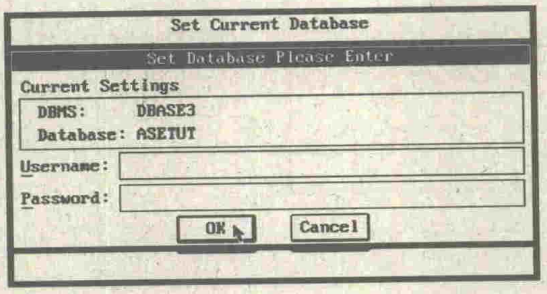
- ขั้นตอนที่ 1 ก็เป็นการ Initialize แบบที่ผ่านมา และให้เป็น Username และ Password ไปก่อนด้วยการกด Enter
- ขั้นตอนที่ 2 ทำการเลือกไดรเวอร์ของ DBMS อันที่เราจะต้องการใช้งาน ซึ่งจะต้องให้ตรงกับฟอร์แมตของข้อมูลดาต้าเบสที่สร้างขึ้นด้วย ในที่นี้ให้เลือกใช้ dBASE3 ดังรูปที่ 6 โดยใช้คำสั่ง

```
Command: ASESETDBMS
```

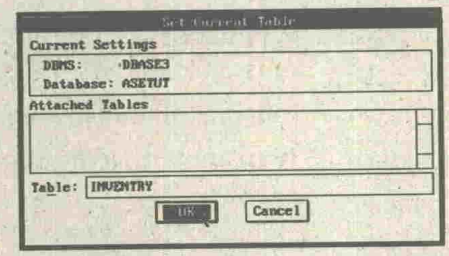
- ขั้นตอนที่ 3 จะเป็นการตั้งชื่อดาต้าเบสที่ต้องการดึงมาใช้งาน ซึ่งจะเป็นชื่อสมมติที่ท่านเคยตั้งชื่อนี้ไว้ขณะอยู่บนดอส (คงจำได้นะครับ) ให้พิมพ์คำว่า "ASETUT" ลงไปในไดอะล็อก-



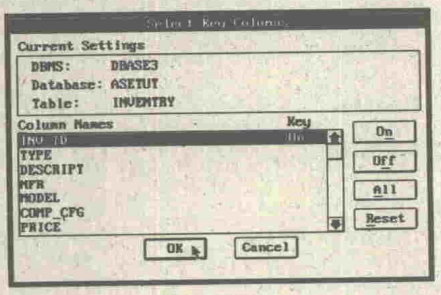
รูปที่ 7 ไดอะล็อกบ็อกซ์ที่ให้เราป้อนชื่อดาต้าเบสที่ต้องการจะใช้งาน



รูปที่ 8 แสดงการปกป้องข้อมูลด้วยการป้อน Password



รูปที่ 9 แสดงการตั้งค่า Current Table ที่ต้องการจะใช้งาน



รูปที่ 10 แสดงการเลือกคีย์อ้างอิงที่ต้องการให้มีสถานะเป็น "ON"

บ็อกซ์ (รูปที่ 7) ซึ่งจะเรียกขึ้นมาโดยพิมพ์คำสั่ง

```
Command: ASESETDB
```

● ขั้นตอนที่ 4 ถัดจากขั้นตอนที่ 3 ออโต้แคดจะรับค่าดาต้าเบสเข้ามา ซึ่งถ้าเรายังไม่ได้ผ่าน Password ในขั้นตอนแรกหรือป้อนชื่อดาต้าเบสตัวใหม่ โปรแกรมจะทำการให้ท่านป้อนชื่อผู้ใช้ (Username) และรหัสผ่าน (Password) เพื่อการปกป้องข้อมูลของท่าน ซึ่งท่านอาจจะป้อนหรือไม่ป้อนก็ได้ ในที่นี้ขอให้ Enter ผ่านไปก่อน (รูปที่ 8)

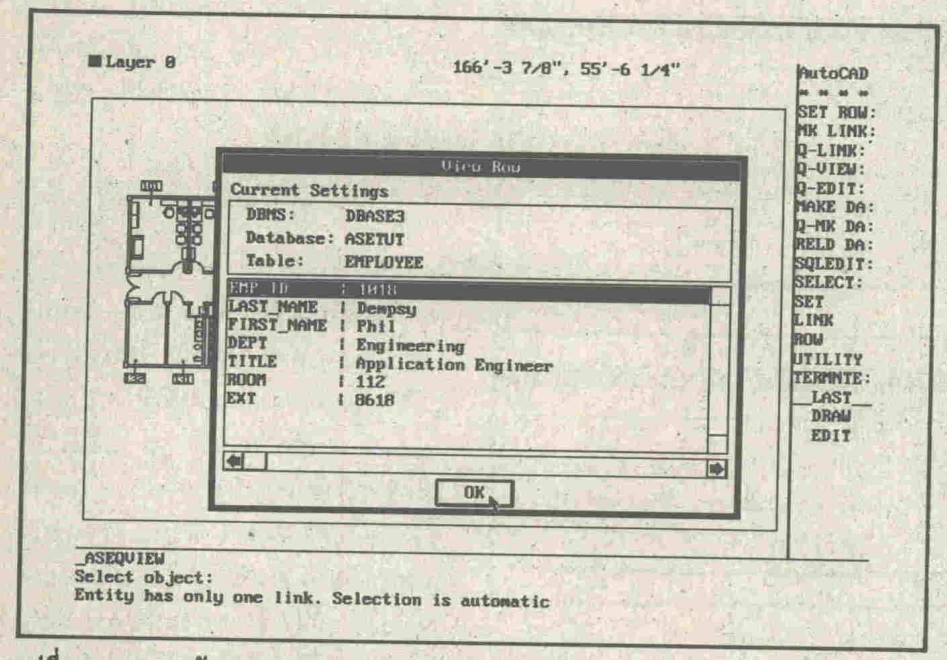
● ขั้นตอนที่ 5 หลังจากที่อ้างอิงชื่อดาต้าเบส "ASETUT" ไปแล้วก็เอาเป็นว่าออโต้แคดเข้าใจแล้วว่าท่านกำลังจะใช้งานบน Tables ในกลุ่มไหน ซึ่งก็คงเป็นหน้าที่ของท่านที่จะไปเลือกใช้ Table นั้น ๆ มาใช้งาน (ถ้าโหลดเข้ามาแล้วก็จะถูกแสดงด้วย) ให้พิมพ์คำสั่งลงไป (รูปที่ 9) แล้วป้อนชื่อ "INVENTRY" และทำการ Enter เพื่อรับค่าเข้าไป

```
Command: ASESETTABLE
```

● ขั้นตอนที่ 6 เป็นขั้นตอนที่ต่อเนื่องจากขั้นตอนที่ 5 เมื่อออโต้แคดรับรู้ว่าเรากำลังดึง Table ใดมาใช้งานแล้วโปรแกรมจะถามต่อถึง Column Names (หรือ Filed ใน dBASE) เพื่อใช้เป็นคีย์อ้างอิง ในการเรียกใช้ข้อมูลที่อยู่ในตารางตัวอื่น ๆ โดยใช้อุปกรณ์ชี้ของเรา Pick ลงไปใน Column Names ที่ต้องการก็จะมีสถานะเป็น "ON" ในที่นี้ต้องการให้ INV_ID มีสถานะเป็น *"ON"* ดังรูปที่ 8 (ถ้าขั้นตอนนี้เคยทำมาแล้ว ออโต้แคดจะข้ามขั้นตอนนี้ไปเลย) หลักในการเลือกคอลัมน์ที่ใช้เป็นคีย์อ้างอิงนี้ เราจะมี *หลักการอยู่ว่า*ควรจะเลือกคอลัมน์ที่มีค่าข้อมูลในส่วนนั้นจะต้องไม่ซ้ำกัน เพื่อความถูกต้องแม่นยำในการเข้าถึงข้อมูลที่ท่านจะต้องตระหนักถึงเรื่องนี้ด้วยนะครับ

● ขั้นตอนที่ 7 ใช้คำสั่ง ASESETABLE อีกครั้ง แต่เปลี่ยนชื่อ Table ไปเป็น "Computer" และ "Employee" และกำหนดคีย์อ้างอิงให้เป็น "COM-CFG" และ "EMP_ID" ตามลำดับ

จากนี้ไปหลังจากที่ท่านเซฟข้อมูลไปแล้วในการ Initialize ครั้งต่อไปออโต้แคดจะข้ามขั้นตอนเหล่านี้ไปทั้งหมดยกเว้น Password เท่านั้น ซึ่งหลังจากนี้แล้วออโต้แคดก็จะข้ามไปสู่การใช้คำสั่งอื่นต่อไปอีกทันทีเลย แต่ขอบอกตรงนี้ได้เลยว่าในขั้นตอนตั้งแต่ที่ 1 ไปจน 7 ถ้าหากท่านใช้ drawing



รูปที่ 11 แสดงดาต้าเบสของ Entity บนไดอะล็อกบ็อกซ์

file "ASETUT.DWG" ที่ได้รับการลิงค์บ้างบางส่วนไปแล้ว จึงทำให้ขั้นตอนบางขั้นตอนจะมีการข้ามไป เช่น ขั้นตอนที่ 4 และ 6 ก็ไม่ต้องแปลกใจนะครับ ไม่ได้หมายความว่าท่านทำผิด เพราะขั้นตอนบางขั้นตอนจะกำหนดค่าเฉพาะในครั้งแรกเท่านั้น แต่พอกลับมาเริ่มใหม่ก็จะข้ามไปเลย ถ้าท่านเปลี่ยนใช้ชื่อดาต้าเบสเป็นชื่ออื่นที่ไม่ใช่ชื่อ "ASETUT" ออโต้แคดก็จะทำตามทุกขั้นตอน ตั้งแต่ขั้นตอนที่ 1 ไปจนถึงขั้นตอนที่ 7 อย่างไม่ผิดเพี้ยนเลยทีเดียว แต่ก็อย่าลืมไปคอนฟิกูเรชันดาต้าเบสด้วยนะครับ

ลองใช้งาน ASE

ผมเชื่อว่าหลายท่านคงจะคันไม้คันมือเป็นที่สุด อยากจะรู้ว่าเจ้า ASE ที่ว่านี้มันจะสำแดงฤทธิ์ได้สักแค่ไหน ต้องใจเย็นไว้ก่อนนะครับ เพราะสิ่งที่จะต้องพูดกันมีมาก แต่เนื้อที่กระดาษมันมีจำกัด ก็ขอยกยอดไปเหมาจบลงในฉบับหน้าก็แล้วกัน แต่ไหน ๆ เครื่องกำลังร้อนอยู่ จะให้ดับลงมันคงจะยาก ก็ให้ท่านชิมลางดูบ้างจะได้รู้ว่ามันวิเศษแค่ไหน...

เอาเป็นว่าตอนนี้ Entity ของเราบน drawing บางส่วนให้ถูกลิงค์กับดาต้าเบสเรียบร้อยแล้ว และท่านเองก็ต้องการไปดูข้อมูลส่วนที่เป็นดาต้าเบสในแต่ละ Entity ที่ลิงค์ไปว่ามันประกอบไปด้วยข้อมูลอะไรบ้าง ก็สามารถกระทำได้โดยพิมพ์คำสั่งดังนี้

```
Command: ASEQVIEW
```

แล้วโปรแกรมจะถามต่อว่า

```
Select objects:
```

ให้เราไป Pick ตรง Entity ใด ๆ ก็ได้ ซึ่งถ้าไม่ตรงกับ Entity ที่ไม่ได้ลิงค์ก็จะแจ้งให้ทราบว่า

```
No link found
```

แต่ถ้า Pick ตรง Entity ที่มีการลิงค์กับดาต้าเบสไปแล้วก็จะเกิดไดอะล็อกบ็อกซ์ดังตัวอย่างในรูปที่ 11 ที่แสดงถึงดาต้าเบสที่ตัวมันเองลิงค์อยู่

ก่อนจาก

ก็คงพอจะมองออกนะครับว่ามันมีข้อมูลดาต้าเบสอยู่ภายในจริง ๆ ไม่ได้ยกเมฆหรือแหกเนตรของท่านแต่ประการใดเลย และเรื่องราวของ ASE ยังมีอีกมาก ยังไม่จบลงง่ายนะครับ เอาไว้ฉบับหน้าก็แล้วกัน สำหรับท่านที่ติดขัดและมีปัญหาจะโทรศัพท์มาพูดคุยกันก็ได้ที่เบอร์ 3773480 ต่อ 1337 หรือติดต่อแบบฮอตไลน์เลยก็ได้ที่เบอร์ 4881068 โทรมาเลยนะครับ สวัสดีครับ...